

Краткий обзор HTML и CSS

HTML (HyperText Markup Language) - наиболее распространенный язык разметки web-документов. История HTML берёт своё начало с 1986 года, когда Международной организацией по стандартизации был принят стандарт "ISO 8879:1986 Information processing—Text and office systems—Standard Generalized Markup Language (SGML)". SGML (Standard Generalized Markup Language) - метаязык, предназначенный для создания языков разметки документов.

Первоначальная версия HTML была разработана ученым Европейского совета по ядерным исследованиям Тимом Бёрнерсом Ли в 1986-1991 на основе SGML и предназначалась для форматирования научных документов. Язык был построен на наборе структурных и семантических элементов - дескрипторов(или тегов). Вплоть до 1994 года язык использовался лишь для структурной разметки документов, в этом же году создаётся организация W3C (World Wide Web Consortium), возглавляемая Тимом Бёрнерсом Ли, целью которой является контроль и внедрение технологических стандартов web. Уже в ноябре 1995 года выходит стандарт HTML 2.0, главным дополнением новой версии стало введение пользовательских элементов управления. В 1997 году вышел стандарт HTML 3.2, расширяющий возможности языка, позволяя использовать таблицы, апплеты, надстрочный и подстрочный текст, также были введены новые теги. Главным новшеством версии стала поддержка CSS (Cascade Style Sheets) - технологии, созданной в 1996 году, предназначенной для описания внешнего вида документа, и, соответственно, для разделения логической структуры документа и описания его внешнего вида. Стандарт CSS1, соответствующий этому периоду, был также принят организацией W3C, разработка велась Хоком Виум Ли и Бертом Босом - специалистами в области информатики. CSS представляет собой формальный язык, на котором пишется перечень правил CSS, помещаемый в так называемые "таблицы стилей". Таблицы могут быть расположены в отдельном файле или внедряются

непосредственно в html-файлы. Стандарты CSS поделены на уровни. CSS1 на данный момент устарел и вышел из использования. Стандарт HTML 4.0, выпущенный в том же 1997 году уже был лишён описательных элементов, утративших свою актуальность с введением CSS, но наиболее важным аспектом нового стандарта стало появление объектной модели страницы (Document Object Model или DOM). DOM является программным интерфейсом для HTML, XML и SVG документов. Это обеспечивает структурированное представление документа (дерева), и определяет способ, по которому структура может быть доступна для программы, для изменения структуры документа, его стиля и содержания. Введение DOM стало прорывом в области web-дизайна, т.к. стало возможным манипулирование элементами страницы web-документа при помощи скриптовых языков.

12 мая 1998 года была принята рекомендация CSS2, основные нововведения - блочная вёрстка, звуковые таблицы стилей, возможность адаптировать вид страницы в зависимости от используемого носителя, управление отображением контента, выходящего за границы элементов и т.д. В 1999 году выходит последняя редакция HTML 4.01, расширяющая возможности, уже имеющиеся в предыдущей версии, также были разработаны руководства по обеспечению доступности документов для людей с ограниченными возможностями. 7 июня 2011 была принята рекомендация CSS2.1. Были исключены некоторые недоработки версии 2, исправлен ряд ошибок, дополнены существующие возможности, введены визуальные эффекты. В настоящее время идёт разработка CSS2.2, черновой вариант датируется 12 апреля 2016 года. Вместе с этим ведётся активная работа над каскадными таблицами третьего поколения - самой масштабной редакцией. Стандарт позволит использовать анимированные элементы и прочие динамические структуры без применения JS. Последний уровень на текущий момент - CSS4, состоящий из множества модулей, и существующий только в виде черновиков.

Планируется введение множества дополнительных селекторов, введение переменных непосредственно в CSS, улучшение средств для работы с изображениями.

В 2014 году был создан новый стандарт HTML5. Он представляет из себя продвинутую версию предыдущего стандарта, со множеством новых возможностей:

- поддержка аудио и видео:
`<audio>` `<video>`;
- рисование с помощью элемента `canvas`:
`<canvas>`;
- новые семантические элементы, атрибуты формы, сокращенные формы тегов:
 - `<article>` - определяет статью в документе;
 - `<aside>` - определяет контент, находящийся в стороне от основного контента страницы;
 - `<bdi>` - изолирует текст от изменения направления вывода;
 - `<dialog>` - определяет диалоговое окно;
 - `<nav>` - определяет набор ссылок навигации.

HTML 5.1 – стандарт принятый в 2016 году, не обладал столь значительными изменениями, некоторые из них:

- атрибуты `picture` и `srcset` – позволяют выбирать изображения под адаптивную верстку;
- события `event-source-error`, `event-track-error`, `event-track-load` для установки медиа;
- события `HTMLTableCaptionElement`, `HTMLTableSectionElement`, `HTMLTableRowElement` для элементов в таблицах HTML;
- возможность добавления атрибута `nonce` к `script` и `style`, для использования CSP (Content Security Policy).

В 2017 году был представлен стандарт HTML 5.2.

Краткое описание введений:

1. Поддержка модульного JavaScript.
2. Элемент `<dialog>` - определяет окно или контейнер, предназначенный для создания всплывающих и модальных окон.
3. Атрибут `nonce` для `<link>`.
4. Элемент `<iframe>`.

Важные теги:

1. `<html></html>` - даёт браузеру информацию о том, что это html документ.

Атрибуты:

- `title` - добавляет всплывающую подсказку на веб-страницу;
- `manifest` - указывает файл манифеста, необходимый для создания оффлайн-приложения;
- `xmlns` - указывает пространство имен для XHTML-документов.

2. `<head></head>` - тег, предназначенный для хранения элементов, не отображаемых в теле документа, за исключением тега `<title></title>`. Как правило это – метатеги, предназначенные для хранения информации предназначенной для браузеров и поисковых систем.

Атрибуты:

- `profile` - указывает адрес профиля метаданных.

3. `<link>` - устанавливает связь с внешним документом вроде файла со стилями или со шрифтами. В отличие от тега `<a>`, тег `<link>` размещается всегда внутри контейнера `<head>` и не создает ссылку.

Атрибуты:

- charset - кодировка связываемого документа;
- href - путь к связываемому файлу;
- media - определяет устройство, для которого следует применять стилевое оформление;
- rel - определяет отношения между текущим документом и файлом, на который делается ссылка;
- sizes - указывает размер иконок для визуального отображения;
- type - MIME-тип данных подключаемого файла.

4. <meta> - определяет метатеги, которые используются для хранения информации предназначенной для браузеров и поисковых систем. Например, механизмы поисковых систем обращаются к метатегам для получения описания сайта, ключевых слов и других данных.

Атрибуты:

- charset - задает кодировку документа;
- content - устанавливает значение атрибута, заданного с помощью name или http-equiv;
- http-equiv - предназначен для конвертирования метатега в заголовок HTTP;
- name - имя метатега, также косвенно устанавливает его предназначение.

5. <noscript></noscript> - контейнер, отображающий свое содержимое, если браузер не поддерживает работу со скриптами или их поддержка отключена пользователем. В остальных случаях браузер игнорирует этот тег и все, что располагается внутри него.

Дополнительных атрибутов не имеет.

6. <script> </script> - предназначен для описания скриптов, может содержать ссылку на программу или ее текст на определенном языке.

Скрипты могут располагаться во внешнем файле и связываться с любым HTML-документом. Такой подход позволяет использовать одни и те же общие функции на многих веб-страницах и ускоряет их загрузку, т.к. внешний файл кэшируется при первой загрузке, и скрипт вызывается быстрее при последующих вызовах.

`<script>` может располагаться в заголовке или теле HTML-документа в неограниченном количестве. В большинстве случаев местоположение скрипта никак не сказывается на работу программы. Однако скрипты, которые должны выполняться в первую очередь, обычно помещают в заголовок документа.

Атрибуты:

- `async` - загружает скрипт асинхронно;
- `defer` - Откладывает выполнение скрипта до тех пор, пока вся страница не будет загружена полностью;
- `language` - Устанавливает язык программирования, на котором написан скрипт;
- `src` - Адрес скрипта из внешнего файла для импорта в текущий документ;
- `type` - Определяет тип содержимого тега `<script>`.

7. `<div></div>` - является блочным элементом и предназначен для выделения фрагмента документа с целью изменения вида содержимого. Как правило, вид блока управляется с помощью стилей. Чтобы не описывать каждый раз стиль внутри тега, можно выделить стиль во внешнюю таблицу стилей, а для тега добавить атрибут `class` или `id` с именем селектора.

Как и при использовании других блочных элементов, содержимое тега `<div>` всегда начинается с новой строки. После него также добавляется перенос строки.

Атрибуты:

- align – задает выравнивание содержимого тега <div>;
- title – добавляет всплывающую подсказку к содержимому;
- class – определяет принадлежность к классу;
- style – возможность задать стили.

8. <a> - является одним из важных элементов HTML и предназначен для создания ссылок. В зависимости от присутствия атрибутов name или href тег <a> устанавливает ссылку или якорь. Якорем называется закладка внутри страницы, которую можно указать в качестве цели ссылки. При использовании ссылки, которая указывает на якорь, происходит переход к закладке внутри веб-страницы.

Для создания ссылки необходимо сообщить браузеру, что является ссылкой, а также указать адрес документа, на который следует сделать ссылку. В качестве значения атрибута href используется адрес документа (URL, Universal Resource Locator, универсальный указатель ресурсов), на который происходит переход. Адрес ссылки может быть абсолютным и относительным. Абсолютные адреса работают везде и всюду независимо от имени сайта или веб-страницы, где прописана ссылка. Относительные ссылки, как следует из их названия, построены относительно текущего документа или корня сайта

Атрибуты:

- accesskey - Активация ссылки с помощью комбинации клавиш;
- coords - Устанавливает координаты активной области;
- download - Предлагает скачать указанный по ссылке файл;
- href - Задает адрес документа, на который следует перейти;
- hreflang - Идентифицирует язык текста по ссылке;
- name - Устанавливает имя якоря внутри документа;
- rel - Отношения между ссылаемым и текущим документами;

- rev - Отношения между текущим и ссылаемым документами;
- shape - Задаёт форму активной области ссылки для изображений;
- tabindex - Определяет последовательность перехода между ссылками при нажатии на кнопку Tab;
- target - Имя окна или фрейма, куда браузер будет загружать документ;
- title - Добавляет всплывающую подсказку к тексту ссылки;
- type - Указывает MIME-тип документа, на который ведёт ссылка.

9. `` - тег, предназначенный для определения строчных элементов документа. В отличие от блочных элементов, таких как `<table>`, `<p>` или `<div>`, с помощью тега `` можно выделить часть информации внутри других тегов и установить для нее свой стиль. Например, внутри абзаца (тега `<p>`) можно изменить цвет и размер первой буквы, если добавить начальный и конечный тег `` и определить для него стиль текста. Чтобы не описывать каждый раз стиль внутри тега, можно выделить стиль во внешнюю таблицу стилей, а для тега добавить атрибут `class` или `id` с именем селектора.

Не имеет дополнительных атрибутов.

10. `<body></body>` - элемент, предназначенный для хранения содержания веб-страницы (контента), отображаемого в окне браузера. Информацию, которую следует выводить в документе, следует располагать именно внутри контейнера `<body>`. К такой информации относится текст, изображения, теги, скрипты JavaScript и т.д.

Тег <body> также применяется для определения цветов ссылок и текста на веб-странице. Подобная практика в HTML 4 осуждается и взамен для указания цветовой схемы рекомендуется использовать стили, применяя их к селектору BODY. Тем не менее, большинство атрибутов до сих пор поддерживается разными браузерами.

Часто тег <body> используется для размещения обработчика событий, например, onload, которое выполняется после того, как документ завершил загрузку в текущее окно или фрейм.

Открывающий и закрывающий теги <body> на веб-странице не являются обязательными, однако хорошим стилем считается их использование, чтобы определить начало и конец HTML-документа.

Атрибуты:

- `alink` - Устанавливает цвет активной ссылки;
- `background` - Задаёт фоновый рисунок на веб-странице;
- `bgcolor` - Цвет фона веб-страницы;
- `bgproperties` - Определяет, прокручивать фон совместно с текстом или нет;
- `bottommargin` - Отступ от нижнего края окна браузера до контента;
- `leftmargin` - Отступ по горизонтали от левого края окна браузера до контента;
- `link` - Цвет ссылок на веб-странице;
- `rightmargin` - Отступ от правого края окна браузера до контента;
- `scroll` - Устанавливает, отображать полосы прокрутки или нет;
- `text` - Цвет текста в документе;
- `topmargin` - Отступ от верхнего края окна браузера до контента;
- `vlink` - Цвет посещённых ссылок.

11.<form></form> - тег устанавливает форму на веб-странице. Форма предназначена для обмена данными между пользователем и

сервером. Область применения форм не ограничена отправкой данных на сервер, с помощью клиентских скриптов можно получить доступ к любому элементу формы, изменять его и применять по своему усмотрению.

Документ может содержать любое количество форм, но одновременно на сервер может быть отправлена только одна форма. По этой причине данные форм должны быть независимы друг от друга.

Атрибуты:

- `accept-charset` - Устанавливает кодировку, в которой сервер может принимать и обрабатывать данные;
- `action` - Адрес программы или документа, который обрабатывает данные формы;
- `autocomplete` - Включает автозаполнение полей формы;
- `enctype` - Способ кодирования данных формы;
- `method` - Метод протокола HTTP;
- `name` - Имя формы;
- `novalidate` - Отменяет встроенную проверку данных формы на корректность ввода;
- `target` - Имя окна или фрейма, куда обработчик будет загружать возвращаемый результат.

12. `<section></section>` - задаёт раздел документа, может применяться для блока новостей, контактной информации, глав текста, вкладок в диалоговом окне и др. Раздел обычно содержит заголовок. Допускается вкладывать один тег `<section>` внутрь другого. Не имеет дополнительных атрибутов.

13. `<title></title>` - Определяет заголовок документа. Элемент `<title>` не является частью документа и не показывается напрямую на веб-

странице. В операционной системе Windows текст заголовка отображается в левом верхнем углу окна браузера. Допускается использовать только один тег `<title>` на документ и размещать его в контейнере `<head>`.

Атрибутов не имеет.

14. `<style></style>` - применяется для определения стилей элементов веб-страницы. Тег `<style>` необходимо использовать внутри контейнера `<head>`. Можно задавать более чем один тег `<style>`.

Атрибуты:

`media` - Определяет устройство вывода, для работы с которым предназначена таблица стилей.

`type` - Сообщает браузеру, какой синтаксис использовать, чтобы правильно интерпретировать стили.

15. `` - предназначен для отображения на веб-странице изображений в графическом формате GIF, JPEG или PNG. Адрес файла с картинкой задаётся через атрибут `src`. Если необходимо, то рисунок можно сделать ссылкой на другой файл, поместив тег `` в контейнер `<a>`. При этом вокруг изображения отображается рамка, которую можно убрать, добавив атрибут `border="0"` в тег ``. Рисунки также могут применяться в качестве карт-изображений, когда картинка содержит активные области, выступающие в качестве ссылок. Такая карта по внешнему виду ничем не отличается от обычного изображения, но при этом оно может быть разбито на невидимые зоны разной формы, где каждая из областей служит ссылкой.

Атрибуты:

- `align` - Определяет как рисунок будет выравниваться по краю и способ обтекания текстом;

- alt - Альтернативный текст для изображения;
- border - Толщина рамки вокруг изображения;
- height - Высота изображения;
- hspace - Горизонтальный отступ от изображения до окружающего контента;
- ismap - Говорит браузеру, что картинка является серверной картой-изображением;
- longdesc - Указывает адрес документа, где содержится аннотация к картинке;
- lowsrc - Адрес изображения низкого качества;
- src - Путь к графическому файлу;
- vspace - Вертикальный отступ от изображения до окружающего контента;
- width - Ширина изображения;
- usemap - Ссылка на тег <map>, содержащий координаты для клиентской карты-изображения.

Список использованных источников

1. Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification // www.w3.org: Консорциум Всемирной паутины. 2016. 12 апр. URL: <https://www.w3.org/TR/2011/REC-CSS2-20110607/> (дата обращения: 22.09.2018).
2. Cascading Style Sheets Level 2 Revision 2 (CSS 2.2) Specification // www.w3.org: Консорциум Всемирной паутины. 2016. 12 апр. URL: <https://www.w3.org/TR/CSS22/> (дата обращения: 22.09.2018).
3. Introduction to CSS3 W3C Working Draft // www.w3.org: Консорциум Всемирной паутины. 2001. 23 мая URL: <https://www.w3.org/TR/2001/WD-css3-roadmap-20010523/> (дата обращения: 22.09.2018).

4. История создания и развития HTML // codingcraft.ru: Школа программирования Coding Craft. 2011. 30 сен. URL: <http://codingcraft.ru/html.php> (дата обращения: 22.09.2018).
5. Справочник HTML // webref.ru: Руководства по веб-технологиям. URL: <https://webref.ru/html> (дата обращения: 22.09.2018).